




# A novel method of spectral clustering in attributed networks by constructing parameter-free affinity matrix

Kamal Berahmand<sup>1</sup> · Mehrnoush Mohammadi<sup>2</sup> · Azadeh Faroughi<sup>3</sup> · Rojjar Pir Mohammadiani<sup>2</sup> 

Received: 10 April 2021 / Revised: 15 August 2021 / Accepted: 24 September 2021 / Published online: 17 November 2021  
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

## Abstract

The most basic and significant issue in complex network analysis is community detection, which is a branch of machine learning. Most current community detection approaches, only consider a network's topology structures, which lose the potential to use node attribute information. In attributed networks, both topological structure and node attributed are important features for community detection. In recent years, the spectral clustering algorithm has received much interest as one of the best performing algorithms in the subcategory of dimensionality reduction. This algorithm applies the eigenvalues of the affinity matrix to map data to low-dimensional space. In the present paper, a new version of the spectral cluster, named Attributed Spectral Clustering (ASC), is applied for attributed graphs that the identified communities have structural cohesiveness and attribute homogeneity. Since the performance of spectral clustering heavily depends on the goodness of the affinity matrix, the ASC algorithm will use the Topological and Attribute Random Walk Affinity Matrix (TARWAM) as a new affinity matrix to calculate the similarity between nodes. TARWAM utilizes the biased random walk to integrate network topology and attribute information. It can improve the similarity degree among the pairs of nodes in the same density region of the attributed network, without the need for parameter tuning. The proposed approach has been compared to other primary and new attributed graph clustering algorithms based on synthetic and real datasets. The experimental results show that the proposed approach is more effective and accurate compared to other state-of-the-art attributed graph clustering techniques.

**Keywords** Complex network · Community detection · Spectral clustering · Attributed network · Affinity matrix

## 1 Introduction

The inherent community structure is ubiquitous in many natural systems and often contains abundant functional information of complex networks, such as the functions of proteins, the patterns of scientific collaboration, the word association in language evolutions, and the emergence of social polarization and echo-chambers [1–3]. Consequently, community detection is fundamental significance for further understanding the complex interplay between network structure and dynamical processes across different fields, ranging from statistical physics, biology, ecology, economics, and social science [4]. Moreover, the process of detecting communities can even contribute to designing more effective data storage systems and improving network capacity [5, 6].

Most of the community detection methods deal only with the structure of networks. There exist a variety of

---

✉ Rojjar Pir Mohammadiani  
r.pirmohamadiani@uok.ac.ir

Kamal Berahmand  
kamal.berahmand@hdr.qut.edu.au

Mehrnoush Mohammadi  
m.mohammadi@eng.uok.ac.ir

Azadeh Faroughi  
A.faroughi@sutec.ac.ir

<sup>1</sup> Department of Science and Engineering, Queensland University of Technology, Brisbane, Australia

<sup>2</sup> Department of Computer Engineering, University of Kurdistan, Sanandaj, Iran

<sup>3</sup> Computer Engineering and IT Department, Shiraz University of Technology, Shiraz, Iran

structure-aware methods in multiple applications [7, 8]. However, with the rapid growth of available information to us, the majority of real-world networks provide attributes describing the properties of nodes in addition to the interconnections. Such network is called as attributed networks and sometimes node attributes are important as the topological structure information. Methods that only consider structure or only attributes, lose some of the available information in a network. Therefore, many algorithms have been proposed to fuse the structure and nodes attribute of a network to detect the communities [9–12].

Some of these algorithms combine structure and attributes before the community detection process [12]. These algorithms first build a similarity matrix based on structure and attribute information and then use this matrix in classical community detection algorithms. According to how structure and attributes information are combined, there are different algorithms like weight-based, node-augmented graph-based, and embedding-based algorithms [12]. These algorithms do not need special software implementation and they just require preprocessing to build the similarity matrix. However, in weight-based algorithms, there is a need to tune hyper-parameters to control the balance between structure and attributes, which is challenging according to different problems. The node-augmented graph-based algorithms don't need any parameter tuning, however, in these algorithms, new nodes and edges are added to the graph which leads to enlargement and increasing complexity in large-scale attributed graphs. For embedding-based algorithms, usually deep learning algorithms are used which improve the accuracy of these algorithms, however the complexity of these algorithms are increased.

For community detection, spectral clustering has attracted a lot of attention in recent years [13, 14]. This method partitions nodes of a graph into groups with a spectral embedding map and usually outperforms traditional clustering algorithms like k-means in dealing with non-convex structures [15]. In spectral clustering, an embedding vector of nodes is constructed in which it maps the nodes of a graph to the k-dimensional points in Euclidean space. For this work, k eigenvectors of the graph's Laplacian matrix are selected and these vectors are a new representation of nodes [16]. After extracting the vectors, a k-way partition algorithm is applied to find the k clusters of nodes. Most of the time, k-means clustering is the algorithm used in this part. Spectral clustering algorithms rely on the analysis of a similarity matrix. Hence, defining a suitable matrix has a high

impact on improving the performance of spectral clustering [17–19]. The input matrix of spectral clustering methods can be adjacency matrix [20], the standard Laplacian matrix [21], the normalized Laplacian matrix [22], modularity matrix [23], and the correlation matrix [24].

However, most spectral clustering algorithms only use the network's structural information and ignore the attributes. Therefore, it is expected that this method's accuracy will not be very high.

In the present paper, we propose a modified version of spectral clustering, called Attributed Spectral Clustering. In order to overcome the challenges of the previous spectral clustering algorithm, we build a new affinity matrix based on both the network structure and attribute information. For building the affinity matrix, first we assign the weights to the edges of a graph based on the similarity of the attributes of nodes. These weights are defined based on cosine similarities of node attributes. Then, we use the biased random walk to find the similarity matrix based on both structure and attributes, where the probability of jumping between nodes is obtained according to the weight of the edge. Hence, nodes with more similarity-based on attributes have a higher probability of walking. Unlike the other fusion algorithms, we don't need to define control parameters to combine structure and attribute. Also, extra nodes and edges are not added to the original network which makes the proposed method appropriate to apply on a large-scale network.

Our main contributions are summarized as follows.

- We propose a new spectral clustering method to detect communities in attributed networks, where an affinity matrix is built based on the information of both network structure and nodes attribute.
- To create the affinity matrix, we leverage a biased Random Walk method. The biased property of this method is obtained by defining different probabilities between nodes in a network structure based on the node's attributes similarities. Therefore, for integrating the attributes and structure information of the network, there is no need for parameter tuning.
- Extensive experiments on different types of synthetic datasets and real-world attributed graph datasets show that our proposed algorithm significantly outperforms five state-of-the-art methods, which are appropriate for the attributed network

This paper is organized as follows. In Sect. 2 we present related work on defining various similarity matrices for spectral clustering and also different approaches to fuse

structure and attribute. In Sect. 3 we describe the algorithm of this paper. In Sect. 4 we report the experiments and results, and finally, in Sect. 5 we conclude this study.

## 2 Related work

Spectral clustering algorithms have been successfully applied to community detection. Since the construction of an excellent similarity matrix is a key to spectral clustering, many methods try to construct an ideal one to obtain a better clustering performance. Zhang and You [25] used the random walk method to construct the similarity matrix. With this approach, they have found similarities between points and their neighbors. The drawback of this method is manually setting the threshold of neighboring nodes which affects the stability of clustering. In [26], Shuxia et al. constructed a similarity matrix of nodes by transition probability among nodes. The Markov chain model is used to calculate the transition probability between nodes. Although this method gets a good accuracy in community detection, it needs a lot of time and space to multiply the transition probability matrix. In [27] Fang Hub et al. developed a Node2vec-SC algorithm that combines node2vec and spectral clustering to find communities in complex networks. The similarity matrix is built by calculating the similarity among any two nodes embedding extracted from the node2vec process. Wang et al. [28] proposed a community detection algorithm based on topology potential and spectral clustering. This algorithm constructs the normalized Laplacian matrix with nodes' topology potential. The topology potential describes the interaction and association among nodes of the network and gives rich structural information of the network. The authors of [29] proposed a method that builds a proximity matrix based on magnitude of the linear coefficients as the similarity values. These linear coefficients are extracted from representation of node as a sparse linear combination of all other nodes in the same network.

The similarity matrix of these considered algorithms is defined based on the topologic of the network, while the attributes of nodes are ignored. To address this problem, some algorithms have been proposed which consider both attribute and structure for creating a similarity matrix. One of the categories of these algorithms is weight-based algorithms which change the node attributed network into a weighted network with no attributes. Then any type of community detection method which is suitable for weighted networks could be applied [30]. In this class of methods, first, the structural information (i.e., edges) of the network is stored as a function of the

similarity between nodes and then it is combined linearly with the similarity calculated according to nodes attributes. Edge weights of the graph are usually assigned as  $W_\alpha(v_i, v_j) = \alpha W_S(v_i, v_j) + (1 - \alpha) W_A(v_i, v_j)$ ,  $v_i, v_j \in V$ , where  $W_S$  and  $W_A$  are chosen structural and attributive similarity functions, respectively. The hyper-parameter  $\alpha$  ( $\alpha \in [0, 1]$ ) controls the balance between structure and attributes. collaborative similarity measure (CSM) [31], Attracting Degree and Recommending Degree (AR-Cluster) [32] are examples of these algorithms. Structural attributed graph cluster (SAG-Cluster) [33] measures similarity based on attribute importance in case the pair of disconnected nodes as well as a novel path strategy using classic Basel problem [34] for the indirectly connected nodes. The spectral algorithm based on node convergence degree (SCNCD) [35] defined a node convergence degree measurement by combining structure with node attribute, and then the overlap communities' structure will be gained through the spectral clustering method. Structure convergence degree and attribute convergence degree are combined using the weighted sum method. The weighting factor can be set according to the actual situation. Alinejad et al. [36] have proposed weight modification approach (PWMA) and proposed linear combination approach (PLCA) methods. PWMA [36] takes the original attributed network as input and transforms it into a non-attributed secondary network. In this step similarity measure(s) would be utilized; in this paper, they include Jaccard, cosine, and angular. The second step is to use a mixed-integer linear programming (MILP) method to detect communities in the constructed secondary network. PWMA neither adds nor removes edges from the original network and only weights change throughout this process. The idea behind PLCA [37] is that by using a linear combination of the network topology and attributes, the nodes which have a similarity more than a desired threshold but are structurally far from each other could get closer. PLCA also includes the transformation step to the secondary network but its edge set doesn't necessarily match the one from the source network and edges may be added. CNS [38] used both Coupled Attribute Similarity and Coupled Attribute-to-Structure Similarity extracted respectively from Node Attribute Information and Network Structure Information. Therefore, CNS uses the most information to detect communities. Caiyan Jia et al. [39] proposed a method called k Nearest Neighbor (KNN-enhanced) which adds the (kNN) graph of node attributes to the original network. Then community centers are determined based on the K-rank-D method, and finally, kNN-nearest or kNN-Kmeans is applied to form communities. In [40] a genetic algorithm for detecting a community in attributed

graphs is proposed where a linear combination of structural connectivity and node attributes, is used as a fitness function in the genetic algorithm. In 2020, the Structure-Attribute Similarities Label Propagation (SAS-LP) algorithm was presented by Kamal et al. [41]. In this algorithm, a new version of the LPA algorithm for attributed graphs is proposed. The problem of these weight-based algorithms is manually chosen hyper-parameters which is challenging in different problems [10, 11].

Another category consists of node-augmented graph-based algorithms. Structural attributed (SA-Cluster) [37] is an example of this category that provides an attribute augmented graph, where attributes are added to the original graph as attribute vertices and attribute edges. Then a neighborhood random walk model is applied to unify two similarities based on structure and attributes and a distance matrix is defined. Cheng et al. [42] proposed an algorithm called Inc-Cluster based on the idea of the augmented graph of SA-Cluster, where in this algorithm time complexity is reduced by incrementally updating the random walk distance. Huang et al. [43] leverage a cell-based subspace clustering approach and propose )SCMAG( algorithm, for community detection in multi-valued attributed networks. The random walk is used to calculate the similarity in the attributed augmented graph. Since the size of the augmented graph is larger than the original one, running these methods in large-scale attributed graphs is hard.

Moreover, embedding techniques are used for node-attributed networks to encode both structures and attribute information [44–46]. Then clustering algorithms like k-means are applied to the learning embedded. Le et al. [47] advocated a framework for document networks that combines topic modeling and graph embedding of documents relationships. In order to join these two spaces, a mapping function from the embedded space to the topic space is proposed. Recently, deep learning algorithms are applied for attributed graph clustering. Deep Attributed Network Embedding (DANE) algorithm [44] is developed which uses two autoencoder architectures to learn node embeddings based on graph structure and node attributes respectively, which can capture the high non-linearity information. In ANRL [48] a neighbor enhancement autoencoder to model the node attribute information and also attribute-aware skip-gram model to capture the network structure are designed. Wang et al. [49] proposed a graph attentional autoencoder to combine both graph structure and attribute values to learn embedding representation. Then, self-training clustering

is performed based on the learned representation. Zhang et al. [50] proposed an adaptive algorithm based on graph convolution network (GCN) [51] 52 to get a new representation of the node. This method adaptively selects the appropriate order for graphs with different diversity. The marginalized graph autoencoder (MGAE) algorithm [53] proposed a newly marginalized graph autoencoder to learn representation for graph clustering. Sun et al. [54] proposed a framework consisting of a graph convolutional autoencoder, modularity module, and a self-clustering module to learn graph structure-based representations and clustering-oriented representations together. Luo and Yan [55] proposed an end-to-end network embedding based on high order graph convolutional network to simultaneously optimizes the node embedding learning and community detection. These methods get a good accuracy in community detection, however the complexity of these attributed network embeddings increases as the network grows.

Our work is a weight-based algorithm, where we assign the weight to the edge of a graph-based on attributes similarity. Then we use the biased random walk to find the similarity matrix based on both structure and attributes, where the probability of walking between nodes is obtained according to the weight of the edge. Unlike the other weight-based ones, we don't need to define control parameters to combine structure and attribute. Moreover, contrasting the augment-based methods no nodes or edges are added to the original network. So, this algorithm can be applied in large-scale networks.

### 3 Contribution

Before addressing the algorithm, let us review some definitions and concepts, which are the proposed algorithm's foundations.

#### 3.1 Background and notation

In general, an attributed network is defined by the triple  $G = (V, E, A)$ , where  $V$  denotes the set of nodes,  $E$  denotes the set of edges indicating the existing node relations, and  $A$  implies the set of attribute vectors. The total number of vertices is shown as the value of  $n = |V|$ , the total number of edges is  $m = |E|$ , and  $A$  ( $\text{attr}_1, \text{attr}_2, \text{attr}_3 \dots \text{attr}_n$ ) is associated with nodes in  $V$  and describes their features. An attribute vector's dimension is  $n$ . We concentrate on graphs

with binary (interchangeably, label) attributes on nodes in the present paper.

### 3.2 Traditional spectral clustering review

Among different algorithms proposed to perform community detection, spectral clustering (SC) has been studied by many researchers [56–59]. Spectral clustering is very popular in data mining because of its ability to detect arbitrary shape clusters in data spectrum feature space. The reason is that the change of representation induced by the eigenvectors makes the cluster properties of the initial data set much more evident. The basic spectral clustering algorithm consists of four steps: (1) Constructing the similarity matrix. (2) Obtaining the Degree matrix,  $D$ , and the Laplacian matrix,  $L$ . (3) Computing the  $k$  eigenvectors of  $L$ , using their eigenvalues. (4) Performing the K-means clustering algorithm to obtain the community structure of the network. Although these steps seem simple enough, there are still some challenges that need to be handled. One of the most important challenges is the definition of the similarity matrix. The similarity matrix has a direct impact on the performance of the SC algorithm.

### 3.3 Incorporate the information of both structure and attribute

In order to conduct the task of community detection in the attributed network, two data sources can be used. The network and the set of connections between nodes provide the first source of data, while data about the nodes and their attributes provide the second. With the growing number of rich graph attributes, such as user profiles in social networks and gene annotations in protein interaction networks, it is more important than ever to consider both the structure and attribute data of graphs for detecting high-quality communities.

According to the homophily property of social networks, relationships among nodes with similar attributes are greater than those between nodes with different attributes, and they are more likely to connect in the network [41, 60]. As a result, the attribute information may affect the presence of two nodes in the community. The Topological and Attribute Random Walk Affinity Matrix is presented to calculate the similarity between nodes by fusing structure and attribute information to further increase the efficiency and accuracy of node similarity calculation in the attributed network.

Embedding the information of vertex attribute similarity into a transformed weighted graph  $G_0 = (V, E, W)$  is the first step. In particular, in order to quantify the vertex attribute similarity for  $u_i$  and  $u_j$ , an edge weight  $w(e)$  is assigned for each edge  $e = (u_i, u_j) \in E$ . Accordingly, the vertex attribute information of  $G$  is encoded into the weighted graph  $G_0$  as edge weights. In order to measure the similarity of the pairs of nodes, the well-known cosine similarity of the angle between two node vectors is applied. A reason for selecting cosine similarity is its effectiveness for sparse vectors that consider only non-zero values. The attribute similarity is expressed as Eq. (1) for two nodes  $u_i$  and  $u_j$ , whose attribute vectors are  $A_i = \{a_{i1}, a_{i2}, \dots, a_{it}\}$  and  $A_j = \{a_{j1}, a_{j2}, \dots, a_{jt}\}$ , respectively.

$$ATSIM(u_i, u_j) = \frac{\sum_{d=1}^t A_{id}A_{jd}}{\sqrt{\sum_{d=1}^t (A_{id})^2} \cdot \sqrt{\sum_{d=1}^t (A_{jd})^2}} \quad (1)$$

where  $t$  implies the dimension of an attribute vector.

The second step is to implement a biased random walk on the weighted graph in order to find node similarities based on both network structure and attributes. In random walk approaches, each node has a walker, and each walker will randomly pick a neighbor of the node that currently stands on to localize. The random walk similarity is constructed for a pair of nodes using a special transition probability rule [61–63], which can help capture both the information potential of topological and attribute relationships between nodes. A more general transfer matrix can be used [reference] to describe a weight-biased random walk on a graph. The factor  $p_{ij}$  provides the probability that a walker on node  $u_i$  of the graph can move to node  $u_j$  in a single step, where this probability is based on the edge's weight of each pair of vertices  $u_i$  and  $u_j$ . The appropriate weight for each pair of nodes in the network is considered proportional to the attribute similarity (ATSIM) between the nodes extracted according to Eq. (1).

When a transition probability  $p_{ij} = ATSIM_{ij}$  on each link  $(u_i, u_j)$  is assigned, semi-local information is applied by the Local Random Walk (LRW) algorithm [64] to obtain similarities between nodes. The final formula is defined as Eq. (2) according to the Bias Local Random Walk model:

$$S_{ij}^{BLRW}(\eta) = \sum_{l=1}^{\eta} \frac{d_i}{2|E|} \cdot \frac{ATSIM_{ij}}{\sum_{j \in \Gamma(i)} ATSIM_{ij}}(l) + \frac{d_j}{2|E|} \cdot \frac{ATSIM_{ij}}{\sum_{i \in \Gamma(j)} ATSIM_{ij}}(l) \quad (2)$$



where  $\eta$  implies the number of random walk steps and  $d$  and  $E$  denote the degree of node and number of present links in the network, respectively. The graph diameter is employed for the number of random walking steps in order to travel the graph structure best with random walking.

### 3.4 The proposed Attributed Spectral Clustering (ASC)

In this investigation, we consider an attributed graph  $G = (V, E, A)$  where the number of clusters is  $K$ . The purpose is partitioning the node-set  $V$  into  $K$  disjoint subsets  $v_1, v_2, \dots, v_n$ , where  $V = \bigcup_{i=1}^n v_i$  and  $v_i \cap v_j = \emptyset$  for any  $i \neq j$ . Therefore, the nodes within clusters are densely connected with regard to structure, while the nodes in different clusters are sparsely connected; and the nodes within clusters have low diversity in their attribute values with regard to attribute, while the nodes in different clusters may have diverse attribute values. The main of the attributed graph clustering is to achieve well-connected (structured) clusters while their nodes benefit from homogeneous attribute values (content).

This algorithm has four main steps; the first step is the formation of the affinity matrix, which is of special importance. Because the effectiveness and quality of spectral clustering mainly depend on the input affinity matrix between each pair of nodes. The affinity matrix acts as an input and consists of a quantitative evaluation of each pair of points in the data set regarding its relative similarity. In order to use the spectral clustering algorithm in attributed networks, the affinity matrix must contain information on graph structure and node attributes. (In most of the previous works, for the simultaneous use of structure and attribute information, researchers have used the combination of these two information sources, and also some parameters have been used to adjust them. Tuning these parameters dramatically affects the algorithm's performance and takes the algorithm out of the free-parameter mode.

However, in the present article, the authors intend to introduce a new affinity matrix called Topological and Attribute Random Walk Affinity Matrix, which does not require any parameters to combine structure and attribute information. For this purpose, first, the graph is weighted using the information of the attributes, and the weight of each edge of the graph is obtained by applying Eq. (1). Based on the weighted graph, the similarity between every two nodes is obtained using the random walking algorithm

defined in Eq. (2). Since random walking is applied to the weighted graph, it accurately traverses the graph's structure with bias. The similarity obtained for every two nodes is highly accurate due to the use of attribute information and k-hop neighborhood structural data.

TARWAM is constructed that faithfully reflects the similarity information of structural and attribute among nodes in attributed networks. After obtaining the affinity matrix, in the second step, the Laplacian matrix is calculated by  $L = D - S$ . Where  $D$  is degree matrix, which is a diagonal matrix with  $D_i = \sum_j S_{ij}$  and  $S$  is an affinity matrix between nodes. Spectral clustering should use the eigenvalues of the affinity matrix of the data to reduce dimension before clustering in fewer dimensions. Then in the third step, the set from  $k$  to the smallest eigenvalues is selected by the Eigngap approach. A particular way to estimate the number of  $k$  or (connected components) is eigngap. Here  $k$  is chosen such that all eigenvalues  $\lambda_1, \dots, \lambda_m$  are small and  $\lambda_{m+1}$  is relatively large. Actually, Eigngap calculated the difference between two consecutive eigenvalues. Most stable clustering is generally given by the value  $k$  that maximizes the difference expression [14].

By selecting  $k$  eigenvalues, the Laplacian matrix is transferred to space with smaller dimensions and contains more information. The new transferred space has a better description of the structure and attributes information of each node. In the last step,  $k$ -means clustering is applied to the new space of data with more useful information. The nodes within the clusters obtained in this step have the highest edge density and homogeneous attribute, which will be equivalent to communities in attributed graphs.

The steps of the ASC algorithm are illustrated in Fig. 1. In this figure, the second column shows the steps of affinity matrix formation, which is indicated by implementing the affinity algorithm of local and biased random walking on the weighted matrix, and the third column shows the other steps of the algorithm.

The details of the algorithm are presented in Fig. 1. As shown in the figure, in step (1), structural and attributes information of nodes are considered. In step (2), the weighted matrix will be defined by calculating the weight edge between each node pair  $(n_i; n_j)$  according to Eq. (1). Then the affinity matrix will be produced by Eq. (2). Using the affinity matrix from the previous step, Laplacian will be calculated in step (3), eigenvalues and eigenvectors of the matrix will be computed, and finally, assign points to two or more clusters, based on the new representation.

### 3.5 Pseudocode

**Algorithm 1: The proposed ASC community detection algorithm**

**Input:** network  $G=(V, E, A)$

**Output:** Community structures  $C = \{C_1, \dots, C_k\}$

1. Convert attributed network to weighted network by calculating the weight edge between each node pair  $(ni; nj)$  according to Eq. (1);
  2. Calculate the similarity  $w_{ij}$  between pairwise points  $(ni; nj)$  by Eq. (2), and construct the affinity matrix  $S S \in R^{n \times n}$ ;
  3. Construct the degree matrix of graph  $D$ ,
  4. Compute the normalized graph Laplacian  $L: L = D^{-1/2}WD^{-1/2}$
  5. Compute the first  $k$  eigenvectors  $u_1; \dots; u_k$  of  $L$ ,
  6. Apply  $K$ -means clustering to obtain the final result.
11. **Return** community structures.

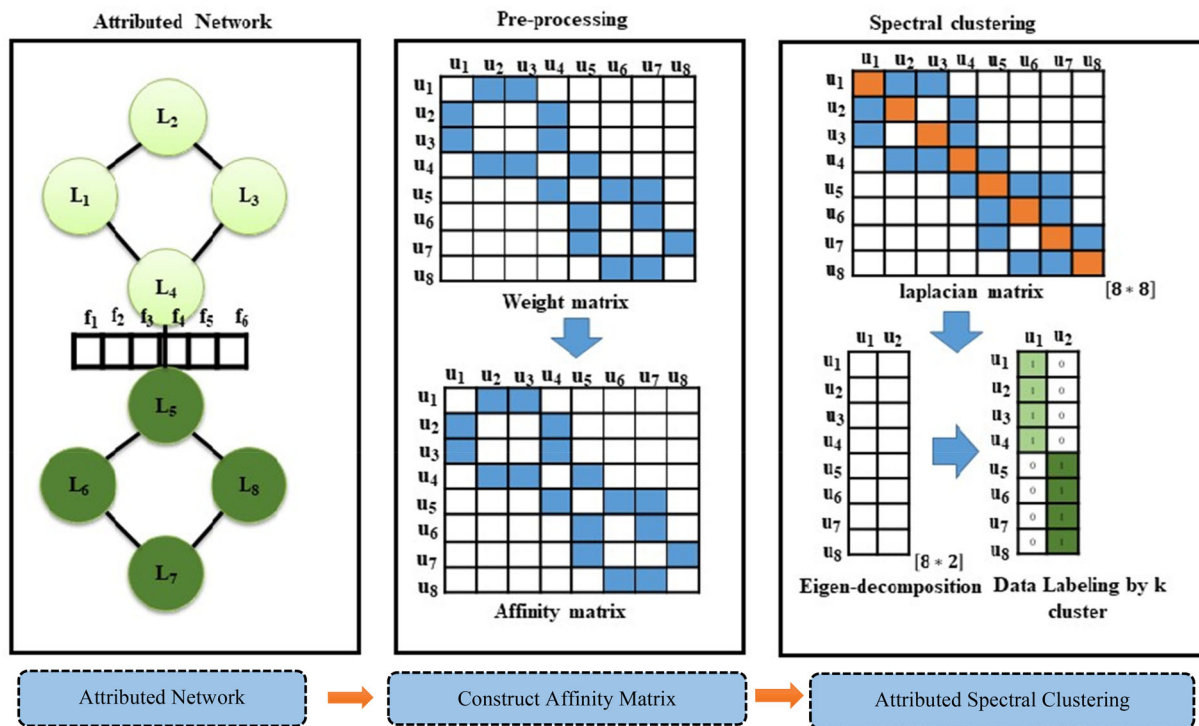


Fig. 1 the steps of the ASC algorithm

### 3.6 Time complexity

We present a complexity analysis of the proposed approach in this section. Assuming that each network has  $N$  nodes,  $|E|$  links. The ASC method consists of three main steps: construct the similarity matrix, compute the first  $k$ -eigenvectors, and  $k$ -means to cluster the normalized matrix  $U$ . The construction of a similarity matrix is divided into two stages: the first is the conversion of the adjacent matrix to a

weighted matrix; because cosine similarity is used, the complexity is equal to  $O(Nk)$  because only the neighbors are considered, where  $d$  represents the average degree of the network; the second stage involves performing a local random walk over a weighted graph with a complexity of  $O(Nk)$ . The first  $k$  eigenvectors from the Laplacian matrix are computed in the second step, which has a complexity of  $O(kN^{2.3676})$ . The  $K$ -means algorithm on dimensional reduction is the final step in performance, with a

complexity of  $O(Ndc)$ , where  $c$  denotes the number of cluster iterations and  $d$  is the dimension of each data set. By combining the results of these analyses, the time complexity of the ASC method is found to be  $O(2Nk + kN^{2.3676} + Ndc) \approx O(kN^{2.3676})$ .

## 4 Experimental evaluation

The ASC experiment findings are discussed in this section. A series of experiments was carried out to evaluate the proposed approach of performance thoroughly. The following is the structure of the organization. Section 4.1 reviews five well-known and state-of-art comparison methods including, SCNCD [35], SA-cluster [37], GA-Net [40], and KNN-enhance [39]. Section 4.2 reviews evaluation metrics. The datasets utilized in the following studies are summarized in Sect. 4.3. In Sect. 4.4, the method's effectiveness is evaluated on different types of synthetic datasets. The synthetic dataset's detailed findings were discussed compared to five state-of-the-art approaches based on four frequently used evaluation metrics. The outcomes of the performance evaluation on real-world datasets are presented in Sect. 4.5. The comparison methods are compiled in the MATLAB programming language and implemented on a computer with an Intel Core i5 processor and 8 GB of RAM.

### 4.1 Comparison methods

In our experiments, the proposed method compares with four well-known and state-of-art methods for discovering communities in the node attributed networks, SCNCD, SA-cluster, GA-Net, and KNN-enhance. The procedure of these methods is summarized as follows:

- SCNCD: It is categorized as weighted-based method in which a weighted linear combination of both structure and attribute controlled by  $\alpha$  parameter, is used to form a similarity matrix. Then, an improved spectral clustering algorithm is applied to this matrix to discover the final communities.
- GA-Net: In this method, the clustering unified distance measure, a linear combination of structural connectivity and node attributes, is used as a fitness function in the genetic algorithm.
- KNN-enhance: Method aim is to reduce the sparsity and noise in the network structure using node attribute enhancement during the community discovering process. To this end, first, the KNN graph of node attributes is added to the original graph. Then the community centers are determined based on the

K-rank-D method. Finally, kNN-nearest or kNN-Kmeans is applied to form final communities.

- SA-cluster: In this method, first, an attribute-augmented graph is formed by combining the structure and attribute of nodes in a unified framework. Then, the neighborhood random walk model is used to obtain a unified pairwise distance. In other words, in this step, the degree of contributions of structural and attributes similarity are automatically learned. Finally, K-means algorithm is adopted to discover the communities based on the pairwise learned distance.

### 4.2 Evaluation metrics

In this paper, the quality of communities generated by different methods is compared by two types of evaluation metrics; quality-based and information recovery-based metrics. In the first type, the quality of discovered communities is evaluated using the basic definition of communities. But information recovery-based metrics are based on the ground truth information of partitions in the networks.

#### 4.2.1 Information recovery-based metrics

Let,  $X$  and  $Y$  be two sets of discovered communities and ground-truth communities, respectively. Then  $x_i$  and  $y_j$  represent the  $i$ th community of these sets.

To evaluate the similarity between these two sets Normalized Mutual Information (NMI) and Rand Index (RI) are used.

**4.2.1.1 Normalized mutual information** Normalized Mutual Information (NMI) [65] is a well-known entropy measure in information theory, which one of its uses is as an evaluation metric to compare the community detection methods. The confusion matrix  $n$  is created to measure of similarity between these two partitions. So that, the values of  $n_{ij}$  is the number of nodes in the  $x_i$  that appear in the  $y_j$ . Then a unified formulation of NMI is defined as:

$$NMI(X, Y) = -2 \times \frac{\sum_{ij} p_{ij} / p_{i+} p_{+j}}{\sum_i p_{i+} \log p_{i+} + \sum_j p_{+j} \log p_{+j}} \quad (3)$$

where,  $p_{ij} = \frac{n_{ij}}{|n|}$ ,  $p_{+j} = \sum_i p_{ij}$  and  $p_{i+} = \sum_j p_{ij}$ . Also,  $|n|$  is the total number of members in the partitioned set. The range of the NMI value is  $[0, 1]$ , higher consistency causes a higher NMI, and  $NMI(X, Y) = 1$  corresponds to being identical to two partitions  $X$  and  $Y$ . Also,  $NMI(X, Y) = 0$  indicates the independence of these partitions.

**4.2.1.2 Rand Index (RI)** RI is the pair counting-based metric [66]. The basic idea behind the RI is that how pairs



of points are clustered. This means the “goodness” of discovered communities is defined as the fraction of a number of the concordant pair nodes in two partitions X and Y as follows:

$$RI(X, Y) = \frac{a + d}{a + b + c + d} \quad (4)$$

where, a: is the set of pairs of nodes that are placed in the same communities in both partitions X and Y. b: is the set of pairs of nodes that are placed into the same communities in partitions X but not in Y. c: is the set of pairs of nodes that are placed into the same communities in partitions Y but not in X. d: is the set of pairs of nodes that are placed in different communities in both partitions X and Y.

#### 4.2.2 Quality-based metrics

The lack of ground-truth communities in many networks has challenged the comparison of community detection methods. So, quality-based metrics are provided to measure the quality of a partitioning of a network based on the definition of networks. In this paper, for the performance assessment of discovered communities, two quality-based metrics; modularity and density are also used.

**4.2.2.1 Modularity** In this metric, the quality of discovered communities is compared to the edges placed within the community with a randomized network [67]. The maximum value of this metric is equal to 1. The closer the value of Q is to 1, the more obvious the community structure. Modularity can be expressed in the following form [3]:

$$Q(X) = \frac{1}{2|E|} \sum_{i,j} (A_{ij} - \frac{k_i k_j}{2|E|}) \delta_{i,j} \quad (5)$$

where  $k_i$  is the degree of node I and  $\delta_{ij}$  is the Kronecker delta function described as:

$$\delta_{ij} = \begin{cases} 1 & v_i \text{ and } v_j \text{ are in the same community.} \\ 0 & \text{Otherwise} \end{cases}$$

**4.2.2.2 Density** This metric measures the density of edges [68] within the cluster and is defined as:

$$den(X) = \frac{1}{E} \sum_{i,j} A(i, j) \cdot \delta_{ij} \quad (6)$$

It ranged into [0, 1]. In other words, the higher values mean more strength of discovered communities.

### 4.3 Datasets

To validate and assess the performance of our algorithm, two classes of datasets are used. The synthetic dataset is computer-generated networks allowing the creation of the ground truth useful to evaluate the similarity between the synthetically generated and the detected communities. The real-world datasets, extracted from real environments, better represent the actual network behavior. The description of these networks is as follows.

#### 4.3.1 Synthetic dataset

LFR-EA is the synthetic network that is generated using the benchmark proposed by Elhadi and Agam [64]. It is an extension of the LFR benchmark of Lancichinetti et al. [69]. The network generator uses two parameters  $\mu$  and  $\nu$ , both ranging in the interval [0.1, 0.8], to control the structure and attribute values, respectively. The mixing parameter  $\mu$  determines the rate of intra and inter-community connections. Low amounts of  $\mu$  give a clear community structure where the intra-cluster link is much more than inter-cluster links. Analogously  $\nu$  is the noise attribute parameter in which low values generate similar features of nodes belonging to the same community. The combination of  $\mu$  and  $\nu$  values produces graphs with a clear to ambiguous structure and/or attributes.

We generated a benchmark of networks consisting of 1000 nodes, named LFREA-1000, to evaluate all aspects of ASC. Different instances of the combination of parameters reported in Table 1 are generated. Since generating networks are the stochastic procedure and different runs may

**Table 1** LFR-EA-1000 parameters setting

Parameter	Value
Number of nodes (N)	1000
Average degree (k)	25
Maximum degree (maxk)	40
Mixing parameter ( $\mu$ )	[0.1;0.8]
Exponent for the community size distribution (t1)	1
Minimum for the community size (minc)	60
Maximum for the community size (maxc)	100
Number of overlapping nodes (om)	0
Number of attributes	4
Attribute's domain cluster assignment (ainf)	1
Attribute range (R)	10
Attribute noise	[0.1;0.9]

1. Mixing parameter  $\mu$ : For the case of varying  $\mu$ , the attribute noise is fixed  $\nu = 0.1$

lead to different resulting partitions, so we average the results over ten runs.

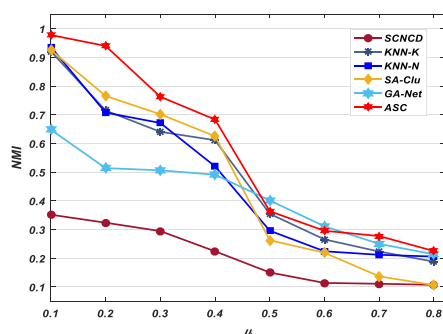
### 4.3.2 Real-world dataset

In addition to synthetic networks, two real-world networks are used for evaluating our experiments; Cora and Cornell. Cora network consists of 2708 nodes representing the machine learning papers, classified into seven classes; Case-Based Reasoning, Genetic Algorithm, Neural Networks, Probabilistic Methods, Reinforcement Learning, or Rule Learning Theory. Each of these papers is associated with 1433-dimensional binary-valued attributes. Also, the citations of these papers are reflected by 5429 edges in the network. The second real-world network is Cornell, one of the four subnetworks gathered from four universities in the WebKB network. In this network, there are 877 websites with 1608 links between them as edges. Each webpage is associated with 1703-dimensional binary-valued attributes (key words of web-pages) and assigned to one of the five communities; course, faculty, student, project, or staff.

### 4.4 Evaluation of synthetic datasets

In the first experiments, we created networks with 1000 nodes by setting 2 numerical attributes for each node. We generated 8 different instances, where the attribute noise parameter ( $\nu$ ) is constant and the mixing parameter ( $\mu$ ) has different values from 0.1 to 0.8. All the nodes in a community, share the same attribute domain values.

Figure 2 shows the NMI result of experiments on the LFR-1000 datasets obtained by the proposed method and compared methods described in the previous section. As shown in this figure, for low values of mixing parameter ( $0.1 \leq \mu \leq 0.4$ ), where the network graph has a clear structure, the proposed method performs better than the other method and achieves higher NMI values compared to the others. SA-Cluster and KNN methods are in the second place of this comparison. For the SCNCD method,



**Fig. 2** NMI comparison of the compared methods on the LFR benchmark networks with  $\mu \in [0.1, 0.8]$  and  $\nu = 0.5$

although optimal alpha is considered, this method is not able to match the ground truth with good NMI values. By increasing the  $\mu$  value, NMI values of all methods are decreasing. When the structure of the graph becomes less clear ( $0.7 \leq \mu \leq 0.8$ ), the proposed method can find the boundary between clear and ambiguous graph structure content better than the other methods. The SA-cluster method makes the biggest drop where the mixing parameter increases.

The other considered metric is RI, where the results are shown in Fig. 3. Each subplot refers to a value of the mixing parameter  $\mu$  ranging from 0.1 to 0.8, with a constant value of attribute noise  $\nu = 0.5$ . Here, the ASC gains the highest rand-index in the most of the mixing parameter values ( $0.2 \leq \mu \leq 0.7$ ) and when  $\mu = 0.1$  and  $\mu = 0.8$  it has the second-highest RI. The SCNCD method has the lowest value in the most of the mixing parameter values. The ranking of the other methods changes in different values of mixing parameters.

In Table 2 the results of the quality metrics (i.e. modularity and density) for LFR-1000 obtained by the proposed method and compared methods are shown. In this table, the highest and the second-highest metrics are marked in italic and bold, respectively. For clear graph structure ( $0.1 \leq \mu \leq 0.2$ ) the ASC has the highest values in both modularity and density. When the mixing parameter increases ( $0.3 \leq \mu \leq 0.8$ ) the ASC still has acceptable results, where the highest or the second highest values of modularity and density belong to the proposed method. Among the other algorithms, SA-cluster and KNN-Nearest have high modularity and density in just some values of mixing parameters. Against the other metrics, the density result of the SCNCD algorithm is high in some values of mixing parameter, however, this metric alone cannot evaluate the algorithms well. According to the results of different evaluation metrics on different networks of LFR-1000, the superiority of the proposed method compared to the other methods is clear. This means the ASC is able to better exploit both the attributes and the structure of the graph on these considered settings.

In the second experiment, we created networks with 1000 nodes by setting 2 numerical attributes for each node. We generated 16 different instances of the combination of  $\mu$  and  $\nu$  parameters. The range of attribute noise values is from 0.1 to 0.8 and the mixing parameter  $\mu$  is constant.

In Fig. 4 the NMI results of the experiments on the graph with two mixing parameter values (0.6 and 0.7) obtained by considering methods are demonstrated. As shown in this figure for  $\mu = 0.6$  the ASC at most of the attribute noise values has the highest NMI. The KNN-nearest loses its efficiency in identifying the communities by increasing the  $\nu$  values. The KNN-kmeans, SA-cluster, and GA-net have lower NMI values in the network with

less attribute noise. When  $\mu = 0.7$  the ASC in graphs with less attribute noise ( $0.1 \leq \nu \leq 0.4$ ) has the second-highest NMI values. By increasing the attribute noise of the network, it outperforms all the other algorithms (except SA-cluster in  $\nu = 0.8$ ). The KNN-Kmeans and KNN-nearest methods which have high NMI values on the graph with less attribute noise, their NMI drop-down severely by increasing the  $\nu$  values. GA-net and SA-cluster are influenced less by increasing the attribute noise, however they have low NMI values in a graph with less attribute compared to the proposed method and KNN algorithms. In both states ( $\mu = 0.6$  and  $0.7$ ) the SCNCD method with an optimal alpha has the lowest NMI and increasing the attribute has a low influence on it.

In Fig. 5 the RI results obtained by experiments on LFR-1000 with  $\mu = 0.6$  are shown where for most of the

attribute noise values ( $\nu = 0.1, 0.3 \leq \nu \leq 0.6, \nu = 0.8$ ) the ASC has the highest, and for some of them ( $\nu = 0.2, \nu = 0.7$ ) it has the second-highest RI. The GA-net algorithm has the second-highest RI in most of the  $\nu$  values. The KNN algorithms have low RI in some values of the  $\nu$  like  $\nu = 0.5$ . Again, SCNCD is the algorithm with the lowest RI in most of the subplots.

The RI values returned by all algorithms on LFR-1000 with  $\mu = 0.7$  are shown in Fig. 6. Also, here the ASC has the highest or the second-highest RI on graphs with different attribute noise. KNN-Kmeans and SCNCD have the lowest RI values in most of the attribute noise values.

The results of quality metrics on LFR-1000 with  $\mu = 0.6$  are reported in Table 3. Here also the best and second-best performance are marked in italic and bold, respectively. According to these metrics, the ASC has the

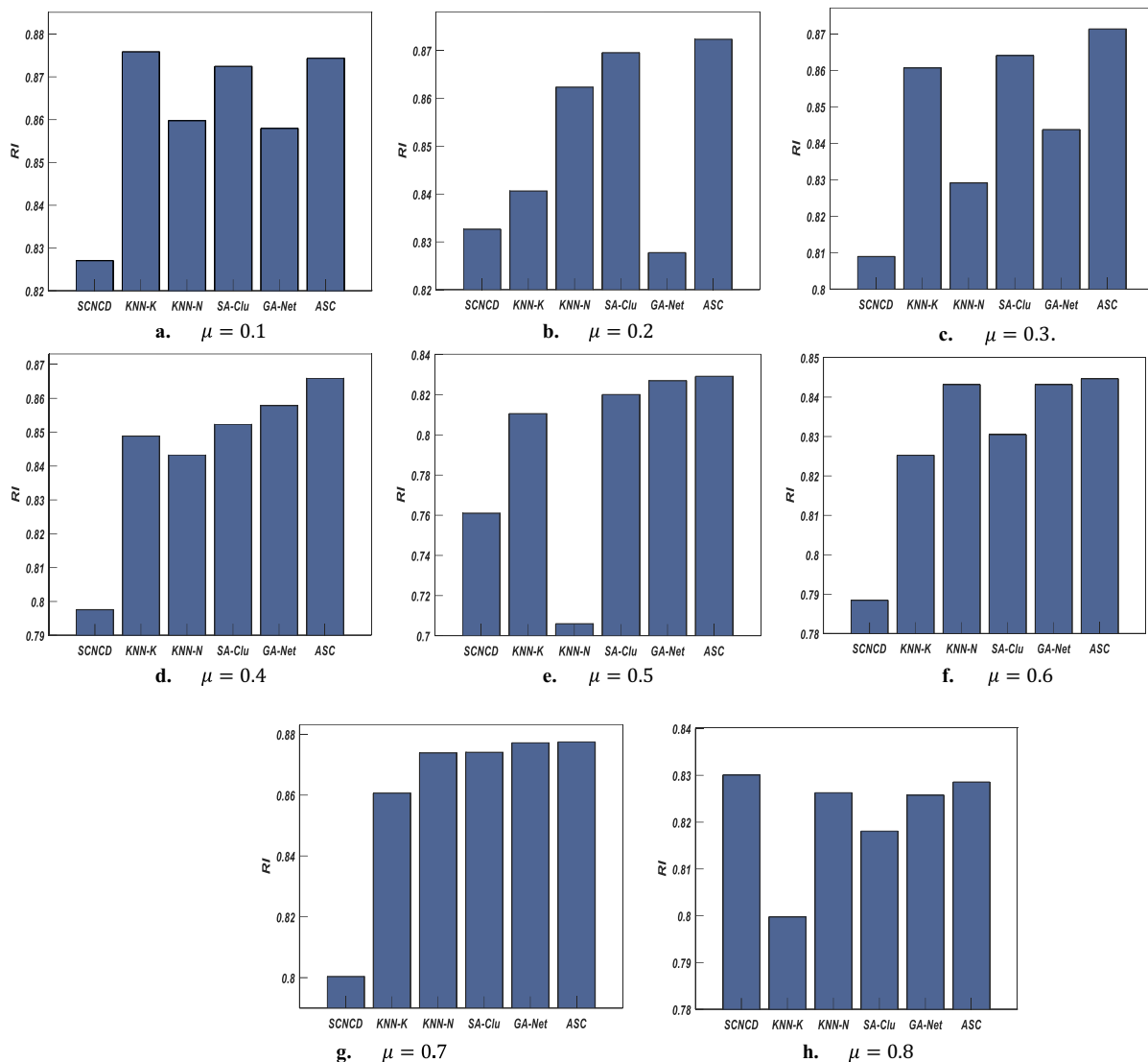


Fig. 3 RI comparison of the compared methods on the LFR benchmark networks with  $\mu \in [0.1, 0.8]$  and  $\nu = 0.5$

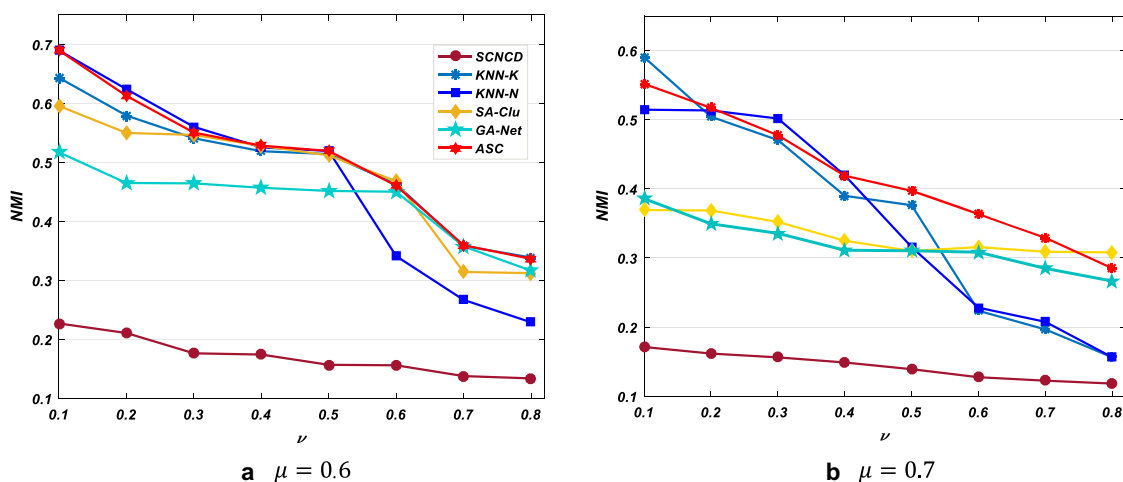
**Table 2** Numerical results of the Modularity and Density comparison of compared methods on the LFR benchmark networks with  $\mu \in [0.10, 0.8]$  and  $\nu = 0.5$

N- $\mu$	Metrics	SCNCD	KNN-K	KNN-N	SA-cluster	GA-Net	ASC
N_0.1	Mod	0.3360	0.6828	<i>0.7004</i>	0.5938	0.1604	<b>0.7088</b>
	Den	0.6916	0.8558	<b>0.8730</b>	0.7775	0.1791	<i>0.8787</i>
N_0.2	Mod	0.4365	<b>0.5107</b>	0.3542	0.4272	0.1496	<i>0.5995</i>
	Den	<b>0.8025</b>	0.7165	0.5224	0.5897	0.1617	<i>0.8116</i>
N_0.3	Mod	0.1580	0.3221	<b>0.4912</b>	<i>0.5626</i>	0.1545	0.4489
	Den	0.6409	0.5090	0.6828	<b>0.8095</b>	0.1733	<i>0.8277</i>
N-0.4	Mod	0.1666	0.3221	<b>0.3542</b>	0.3093	0.1346	<i>0.3549</i>
	Den	0.5938	0.5090	<b>0.5224</b>	0.4686	0.1562	<i>0.5302</i>
N-0.5	Mod	0.1271	0.1664	0.1336	<b>0.1864</b>	0.1393	<i>0.2275</i>
	Den	<b>0.6178</b>	0.3749	0.4671	0.3963	0.1565	<i>0.6242</i>
N-0.6	Mod	0.0649	0.0915	0.0312	<i>0.2084</i>	0.1417	<b>0.1908</b>
	Den	0.5247	0.3768	<b>0.5403</b>	0.3891	0.1601	<i>0.5498</i>
N-0.7	Mod	0.0904	0.0551	0.0211	<b>0.2108</b>	0.1529	<i>0.2306</i>
	Den	<i>0.5542</i>	0.2756	0.4271	0.3685	0.1715	<b>0.4667</b>
N-0.8	Mod	0.0790	0.0352	0.0086	<b>0.1518</b>	0.1274	<i>0.1775</i>
	Den	0.5519	0.3934	<i>0.7151</i>	0.3666	0.1462	<b>0.5615</b>

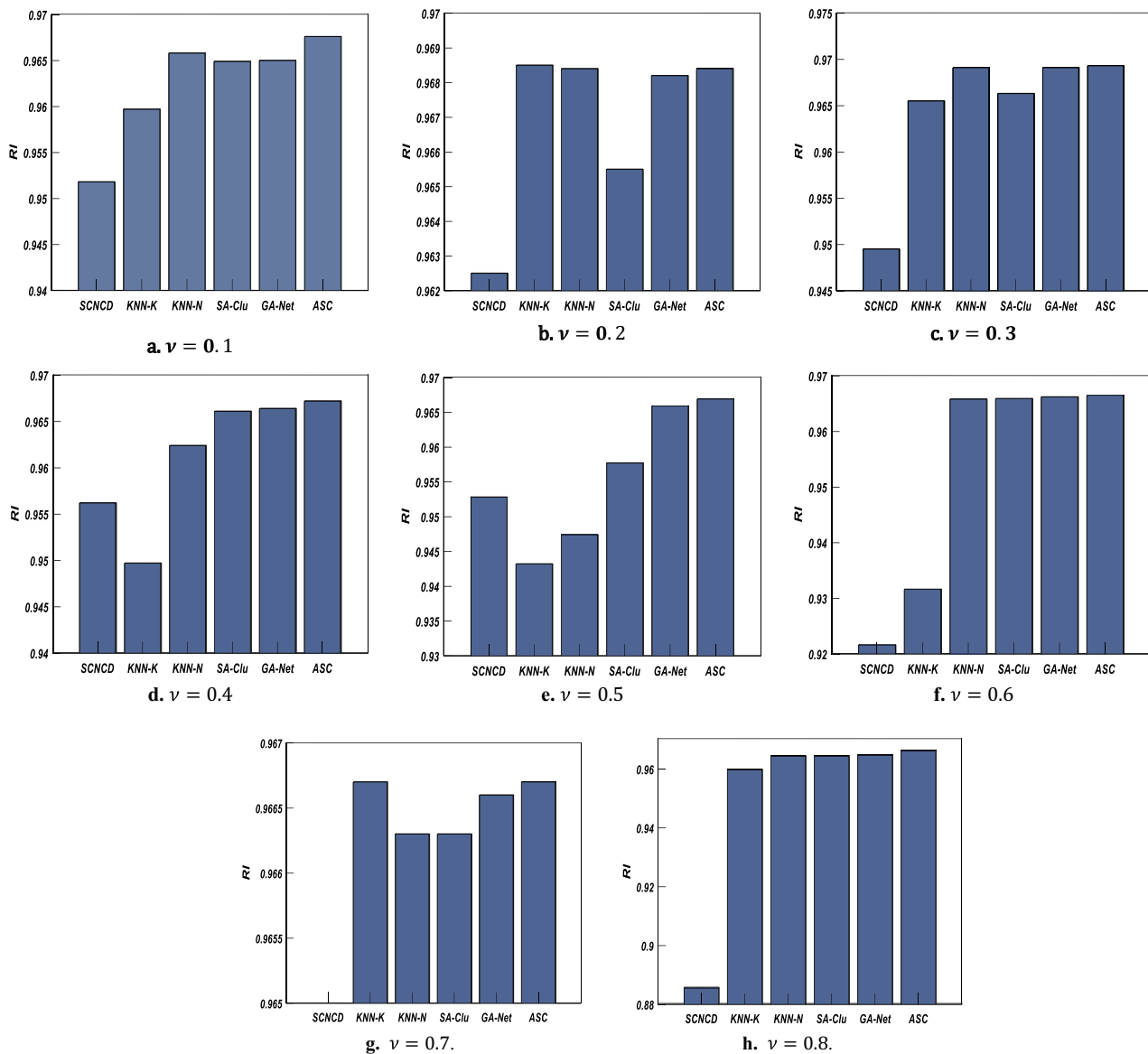
highest density for all the  $\nu$  vales (except  $\nu = 0.7$  where it has the second-highest density). For the modularity metric, the ASC has the highest and second-highest values in different values of  $\nu$ . The KNN methods have acceptable modularity and density when attribute noise is low ( $0.1 \leq \nu \leq 0.3$ ). However, by increasing the  $\nu$  values ( $0.3 \leq \nu \leq 0.8$ ) these methods drop down. SA-Cluster has results comparable to the proposed method in a network with a high attribute value ( $0.3 \leq \nu \leq 0.8$ ). The results of GA-Net and SCNCD are low in both modularity and density for all  $\nu$  values. In Table 4 the results of quality metrics of all algorithms on LFR-1000 with  $\mu = 0.7$  are reported. The ASC for both modularity and density metrics has the highest or the second highest values in different values of  $\nu$ . However, the other algorithms just have a high value in one or

two values of  $\nu$ . Also, the results of the second experiment on LFR-1000 networks in different metrics confirm the superiority of the proposed method in most cases. The ASC has the best or the second-best performance among all comparison algorithms.

In the last experiments on a synthetic dataset, we created LFR networks by setting 5 and 10 numerical node attributes and the different sizes of the network. The parameter settings of these networks are shown in Table 5 we generated 10 different instances of the combination of the number of nodes and number of attributes with constant mixing parameter and attribute noise. Here, we want to evaluate the influence of the number of attributes on the performance of the ASC and comparison ones. The details of the created instances are shown in Table 5.



**Fig. 4** NMI and RI comparison of the compared methods on the LFR benchmark networks with  $\nu \in [0.10, 0.8]$ , **a**  $\mu = 0.6$  and **b**  $\mu = 0.7$



**Fig. 5** RI comparison of the compared methods on the LFR benchmark networks with  $\mu = 0.6$  and, **a–h**  $\nu \in [0.10, 0.8]$

The NMI and RI results obtained by all algorithms on LFR-networks 1–5 with attributes 5 and 10 are reported in Fig. 7. The NMI results of the ASC for all the networks with 10 node attributes are the highest. For the networks with 5 node attributes, the ASC has the highest NMI for graphs with fewer nodes ( $N = 100, 200, 500$ ) and had the second-highest NMI for graphs with more nodes ( $N = 600, 800$ ). GA-Net has results comparable to the proposed method just for networks with fewer nodes. The KNN algorithms and SA-Cluster in networks with small sizes don't have a good performance based on NMI results. The SCNCD is the algorithm with the lowest NMI value for networks with different sizes and different attribute numbers. Based on the RI results, the ASC has the highest RI among all algorithms for all of the network settings except

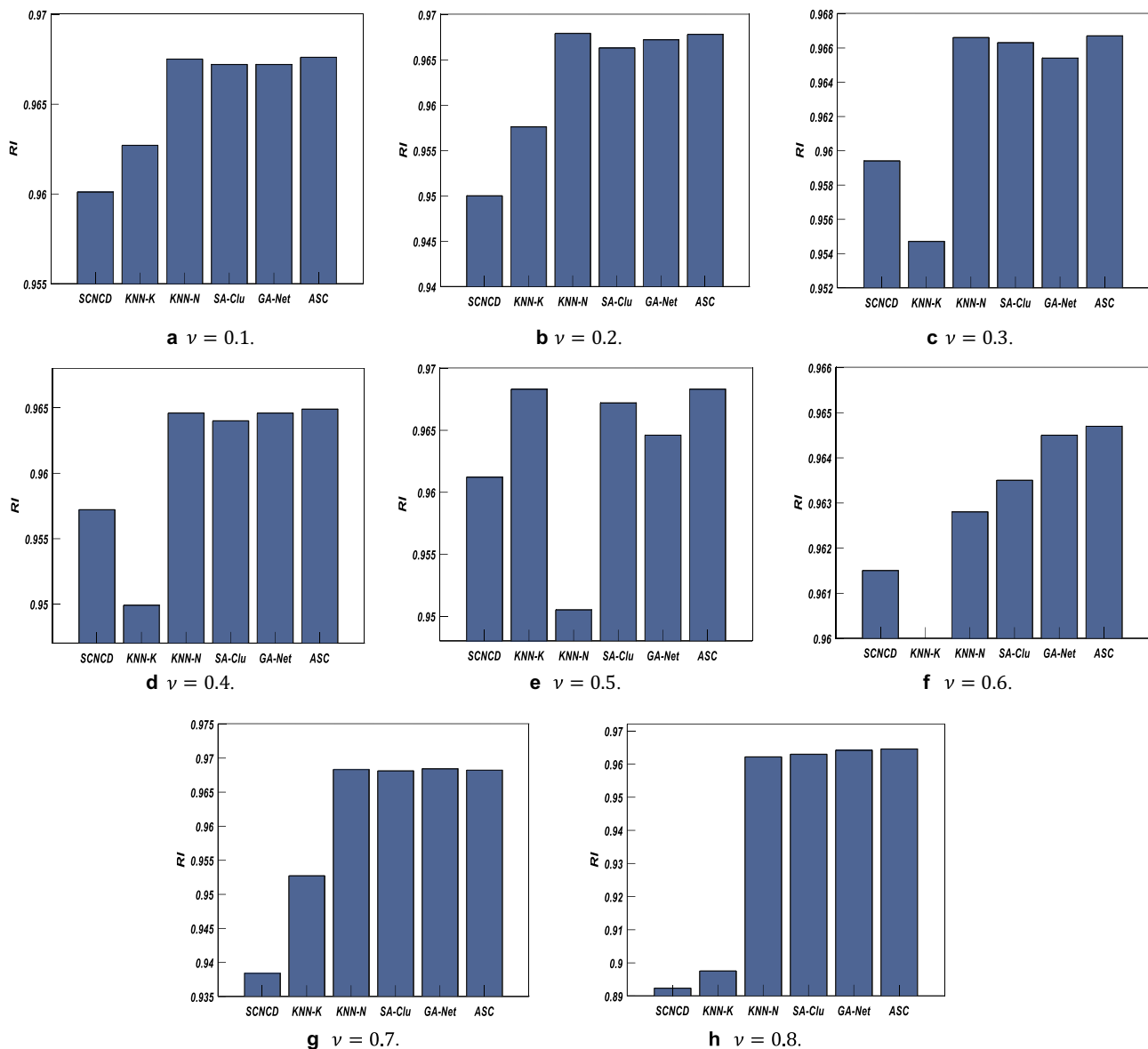
network 2, where it has the second-highest and its RI is so close to the highest one.

In Tables 6 and 7 the results of quality metrics (modularity and density) for networks with 5 and 10 node attributes respectively are reported. (Here also the best and the second-best performance are marked in *italic* and **bold**, respectively.) The superiority of the ASC in both tables is clear compared to the other algorithms. After the proposed method KNN-Nearest has a good performance for most of the different network sizes with node attribute numbers 5 and 10.

#### 4.5 Evaluation of real-world datasets

In this section, to have a better comparison of the performance of the competitors, their performance is also





**Fig. 6** RI comparison of the compared methods on the LFR benchmark networks with  $\mu = 0.7$  and **a–h**  $\nu \in [0.1, 0.8]$

examined on the two real-world networks; Cora and Cornell. Table 8 shows the numerical results in terms of information recovery-based metrics; NMI and RI and also, quality-based metrics; modularity (Mod), and density (Den). In this table, the best and the second-best performance are marked in italic and bold, respectively. The results confirm the superiority of the proposed method in most cases. Therefore, the proposed method has the best or the second-best performance among others in all cases except NMI on the Cora network.

As shown in Table 8, the proposed method is the top-performer in terms of RI and Mod in the Cornell networks, followed by KNN-Nearest in terms of RI and SCNCD in

terms of Den. Also, it has the best performance in terms of NMI and Mod after the GA-Net method. On the other hand, on the Cora network, the proposed method is a top-performer in the RI, Mod, and Den, while it follows the KNN-enhance in terms of MNI. The superiority of the proposed method in terms of quality metrics verifies the strength of the proposed method for discovering communities with high quality. Generally, the results on the real-world networks indicate that the proposed method achieves the best average rank balance on these networks, which implies the superiority of the proposed method.

**Table 3** Numerical results of the modularity and density comparison of compared methods on the LFR benchmark networks with  $\nu \in [0.10.8]$  and  $\mu = 0.6$

N- $\nu$	Metrics	SCNCD	KNN-K	KNN-N	SA-Cluster	GA-Net	ASC
N-0.1	Mod	0.1145	0.3002	<i>0.3154</i>	0.2127	0.1347	<b>0.3101</b>
	Den	0.2104	0.3644	<i>0.3712</i>	0.2471	0.1248	<i>0.3712</i>
N-0.2	Mod	0.0174	0.2611	<b>0.2639</b>	0.2222	0.1167	<i>0.2809</i>
	Den	0.1811	0.3013	<b>0.3138</b>	0.2529	0.1269	<i>0.3543</i>
N-0.3	Mod	0.0104	0.2305	<b>0.2384</b>	0.2308	0.1112	<i>0.2615</i>
	Den	0.0594	0.2661	<b>0.2825</b>	0.2613	0.1191	<b>0.3511</b>
N-0.4	Mod	0.0250	0.1659	0.1731	<i>0.2049</i>	0.1208	<b>0.1952</b>
	Den	0.1672	0.2104	0.2273	<b>0.2483</b>	0.1321	<i>0.2586</i>
N-0.5	Mod	0.0120	0.0755	0.1001	<i>0.1996</i>	0.1176	<b>0.1982</b>
	Den	0.1751	0.1118	0.1510	<b>0.2414</b>	0.1255	<i>0.2493</i>
N-0.6	Mod	0.0116	0.0729	0.0989	<i>0.2081</i>	0.1177	<b>0.2023</b>
	Den	0.0616	0.1153	0.1667	<b>0.2420</b>	0.1274	<i>0.2484</i>
N-0.7	Mod	0.0106	<i>0.3316</i>	0.0718	0.2119	0.1152	<b>0.2842</b>
	Den	0.0816	<i>0.3696</i>	0.1388	0.2451	0.1242	<b>0.3616</b>
N-0.8	Mod	0.0082	0.0128	0.0550	<b>0.1544</b>	0.1173	<i>0.1623</i>
	Den	0.1117	0.1210	0.1503	<b>0.2488</b>	0.1287	<i>0.3682</i>

**Table 4** Numerical results of the modularity and density comparison of compared methods on the LFR benchmark networks with  $\nu \in [0.10.8]$  and  $\mu = 0.7$

N- $\nu$	Metrics	SCNCD	KNN-K	KNN-N	SA-cluster	GA-Net	ASC
N-0.1	Mod	0.0153	<b>0.1767</b>	<i>0.2032</i>	0.1577	0.1144	0.1759
	Den	0.0611	0.2115	<b>0.2448</b>	0.1897	0.1283	<i>0.3175</i>
N-0.2	Mod	0.0120	0.1341	0.1531	<b>0.1623</b>	0.1577	<i>0.1661</i>
	Den	0.2010	0.1701	<i>0.1982</i>	0.1948	0.1897	<b>0.1971</b>
N-0.3	Mod	0.0166	0.1069	0.0951	<b>0.1616</b>	0.1521	<i>0.1620</i>
	Den	0.2719	0.1227	0.1393	0.1943	<b>0.2516</b>	<i>0.3856</i>
N-0.4	Mod	0.0134	0.0828	0.0645	<b>0.1507</b>	0.1169	<i>0.1581</i>
	Den	<b>0.3230</b>	0.1431	0.1116	0.1925	0.1277	<i>0.3442</i>
N-0.5	Mod	0.0093	<i>0.1794</i>	0.0477	0.1420	0.1136	<b>0.1442</b>
	Den	<b>0.2678</b>	0.2317	0.1227	0.1926	0.1227	<i>0.3672</i>
N-0.6	Mod	0.0085	0.0354	0.0261	<b>0.1608</b>	0.1130	<i>0.1639</i>
	Den	<i>0.5030</i>	0.1523	0.0868	0.1968	0.1227	<b>0.4381</b>
N-0.7	Mod	0.0158	0.0227	0.0247	0.1587	0.1103	<i>0.1614</i>
	Den	<i>0.3214</i>	0.1200	0.0918	0.1990	0.1202	<b>0.3185</b>
N-0.8	Mod	0.0089	0.0248	0.0129	<i>0.1578</i>	0.1130	<b>0.1463</b>
	Den	<i>0.3980</i>	0.1358	0.1733	0.1928	0.1234	<b>0.3817</b>

**Table 5** LFR benchmark networks with numerical node attributes,  $\mu = 0.6$  and  $\nu = 0.5$

Networks	K	N	E	NC	N-attr
Net-1	10	100	526/529	3/4	5/10
Net-2	10	200	974/987	5/6	–
Net-3	15	500	3918/3948	19/15	–
Net-4	15	600	4549/4767	19/20	–
Net-5	15	800	6117/6060	27/29	–

### 5 Conclusion

Since community structures frequently disclose both topological and functional relationships between various components of a complex system, community detection is a fundamental and essential problem in network science. In this study, the authors have proposed a new affinity matrix for spectral clustering in the attributed network, which combines structural and attributed information and does

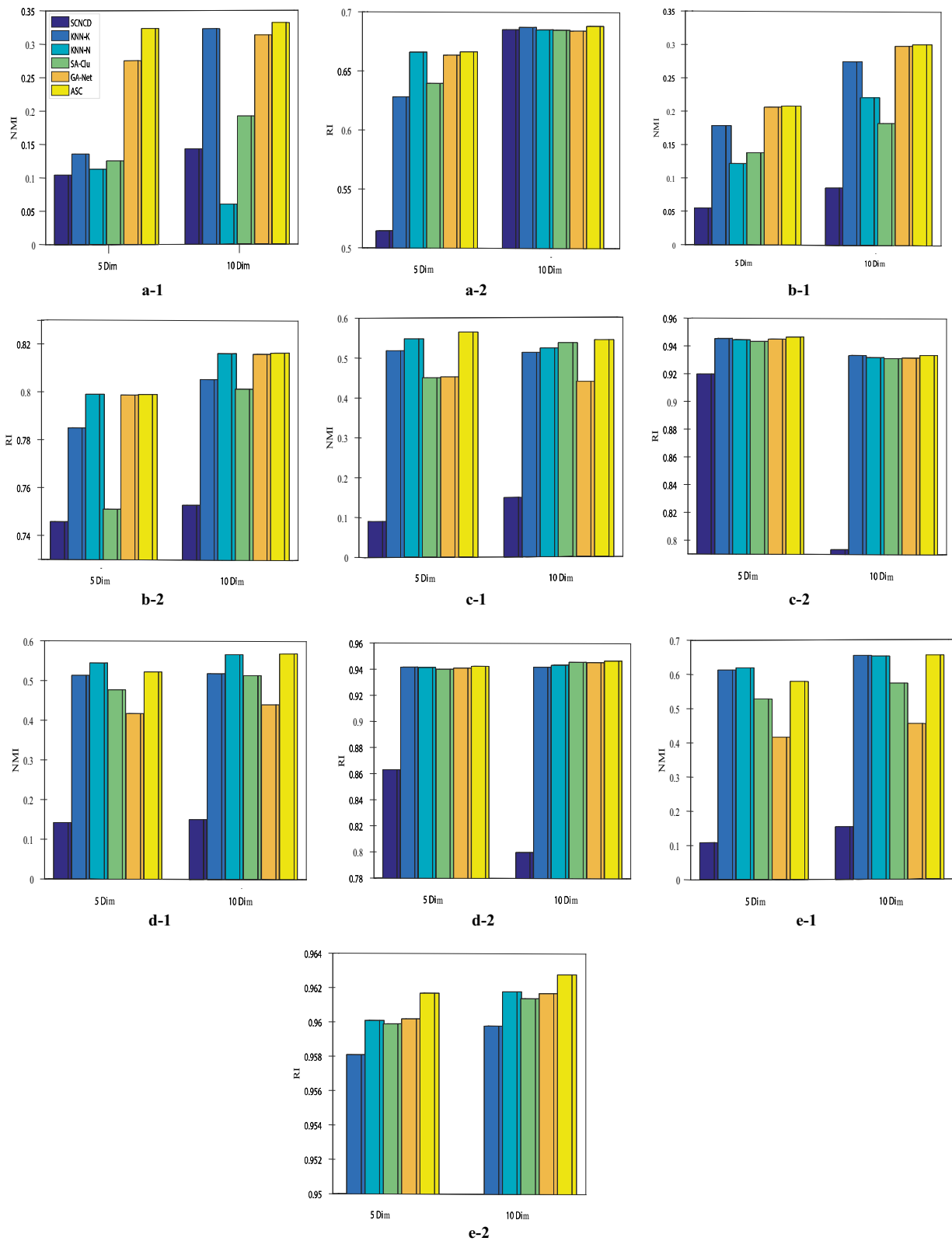


Fig. 7 NMI and RI comparison of the compared methods on the LFR benchmark networks with  $\mu = 0.6$  and  $\nu = 0.5$

**Table 6** The numerical results of comparison methods on the Net-1 according to modularity (Mod) and density (Den). 5 Dim

N-v	Metrics	SCNCD	KNN-K	KNN-N	SA-Cluster	GA-Net	ASC
Net-1	Mod	0.0289	0.0762	0.0175	0.1068	<b>0.1180</b>	0.1474
	Den	0.6445	0.7643	<b>0.8883</b>	0.5722	0.1423	0.8992
Net-2	Mod	0.0406	0.1003	0.0269	<b>0.1831</b>	0.1370	0.1934
	Den	0.5996	0.3974	<b>0.7857</b>	0.4168	0.1517	0.7895
Net-3	Mod	0.0197	0.2175	0.2524	0.2067	0.1047	<b>0.2369</b>
	Den	0.1279	0.3174	<b>0.3369</b>	0.2637	0.1174	0.3698
Net-4	Mod	0.0245	0.2034	0.2245	0.2015	0.1172	<b>0.2093</b>
	Den	0.2653	0.3089	<b>0.3111</b>	0.2565	0.1310	0.3937
Net-5	Mod	0.1160	0.2453	<b>0.2656</b>	0.2225	0.1187	0.2675
	Den	<b>0.3475</b>	0.2903	0.3165	0.2607	0.1314	0.4013

**Table 7** The numerical results of comparison methods on the Net-1 according to modularity (Mod) and density (Den). 10 Dim

N-v	Metrics	SCNCD	KNN-K	KNN-N	SA-cluster	GA-net	ASC
Net-1	Mod	0.0946	<b>0.1464</b>	0.0189	0.0555	0.1099	0.1932
	Den	0.7297	0.5017	0.8773	0.4197	0.1323	<b>0.8336</b>
Net-2	Mod	0.0492	0.1359	0.1117	<b>0.1824</b>	0.1335	0.1994
	Den	<b>0.6282</b>	0.3366	0.4184	0.3982	0.1512	0.7217
Net-3	Mod	0.0413	0.2564	<b>0.2758</b>	0.1690	0.1073	0.2778
	Den	0.2690	0.3518	<b>0.3635</b>	0.2373	0.1211	<b>0.4825</b>
Net-4	Mod	0.0596	0.2532	0.2924	0.1754	0.1087	<b>0.2536</b>
	Den	<b>0.3951</b>	0.3460	0.3566	0.2320	0.1208	0.4053
Net-5	Mod	0.1260	0.2760	<b>0.2926</b>	0.1978	0.1193	0.2983
	Den	0.2469	0.3134	<b>0.3406</b>	0.2355	0.1299	0.3551

**Table 8** Numerical results of the NMI, RI, Mod, and Den comparison of compared methods on the real-world networks

Networks	Metrics	SCNCD	KNN-K	KNN-N	SA-cluster	GA-net	ASC
Cornell	NMI	0.0938	0.0952	0.0265	0.0457	0.2385	<b>0.1228</b>
	RI	0.7176	0.6871	<b>0.7181</b>	0.6111	0.7176	0.7195
	Mod	0.1797	0.2800	0.0768	0.0282	0.4326	<b>0.2858</b>
	Den	<b>0.7497</b>	0.5831	0.4372	0.3055	0.4836	0.8200
Cora	NMI	0.1125	0.3569	<b>0.3469</b>	0.1190	0.1839	0.2136
	RI	0.7603	0.8134	<b>0.8116</b>	0.8042	0.8086	0.8136
	Mod	0.1445	0.4785	0.2305	0.2835	0.4243	0.6489
	Den	0.8101	0.6626	0.5349	0.1233	0.4263	0.8639

not need any parameter. For this purpose, first, the graph is weighted by using the similarity of the attributes of the pairs of nodes, and then the biased random walking is utilized to calculate the similarity between nodes on a weighted graph. The similarity matrix can capture high accuracy similarity between nodes according to structure and attribute. We conducted a comparative experiment on real and artificial networks based on parameters such as modularity, density, NMI, and RI to demonstrate the effectiveness of our proposed method. These studies

explicitly demonstrate the benefits of our proposed method. The experimental findings from tests on various real and artificial networks with varying sizes revealed that our proposed plan outperformed other algorithms. We would attempt to provide a new systematic way to maximize the quality of the new method on vast volumes of data in future studies by providing a procedure that can be applied in parallel and will significantly increase the efficiency. Besides, With the feature selection approach, the node attribute information can be optimally picked to calculate

the similarity between pairs of attribute nodes, and this optimal information can be employed in future works. Furthermore, we plan to study replacing the final step of this algorithm, c-means instead of k-means, to tackle overlap community detection problems.

## References

- Palla, G., et al.: Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **435**(7043), 814–818 (2005)
- Wang, X., et al.: Public discourse and social network echo chambers driven by socio-cognitive biases. *Phys. Rev. X* **10**(4), 041042 (2020)
- Liu, L., et al.: Homogeneity trend on social networks changes evolutionary advantage in competitive information diffusion. *N. J. Phys.* **22**(1), 013019 (2020)
- Fortunato, S.: Community detection in graphs. *Phys. Rep.* **486**(3–5), 75–174 (2010)
- Cai, J., et al.: Enhancing network capacity by weakening community structure in scale-free network. *Futur. Gener. Comput. Syst.* **87**, 765–771 (2018)
- Berahmand, K., Bouyer, A., Vasighi, M.: Community detection in complex networks by detecting and expanding core nodes through extended local similarity of nodes. *IEEE Trans. Comput. Soc. Syst.* **5**(4), 1021–1033 (2018)
- Newman, M.E.: Spectral methods for community detection and graph partitioning. *Phys. Rev. E* **88**(4), 042822 (2013)
- Zhou, L., et al.: An approach for overlapping and hierarchical community detection in social networks based on coalition formation game theory. *Expert Syst. Appl.* **42**(24), 9634–9646 (2015)
- Günemann, S., Boden, B., Seidl, T.: DB-CSC: a density-based approach for subspace clustering in graphs with feature vectors. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, New York (2011)
- Chunaev, P.: Community detection in node-attributed social networks: a survey. *Comput. Sci. Rev.* **37**, 100286 (2020)
- Bothorel, C., et al.: Clustering attributed graphs: models, measures and methods. *Netw. Sci.* **3**, 408–444 (2015)
- Zhou, Y., Cheng, H., Yu, J.X.: Clustering large attributed graphs: an efficient incremental approach. In: *2010 IEEE International Conference on Data Mining*. IEEE (2010)
- White, S., Smyth, P.: A spectral clustering approach to finding communities in graphs. In: *Proceedings of the 2005 SIAM international conference on data mining*. SIAM (2005)
- Zhang, S., Wang, R.-S., Zhang, X.-S.: Identification of overlapping community structure in complex networks using fuzzy c-means clustering. *Physica A* **374**(1), 483–490 (2007)
- Peng, R., Sun, H., Zanetti, L.: Partitioning well-clustered graphs: spectral clustering works! In: *Conference on learning theory*. PMLR (2015)
- Nascimento, M.C., De Carvalho, A.C.: Spectral methods for graph clustering—a survey. *Eur. J. Oper. Res.* **211**(2), 221–231 (2011)
- Zhang, Z., Jordan, M.I.: Multiway spectral clustering: a margin-based perspective. *Stat. Sci.* **23**(3), 383–403 (2008)
- Zass, R., Shashua, A.: A unifying approach to hard and probabilistic clustering. In: *Tenth IEEE International Conference on Computer Vision (ICCV'05) vol. 1*. IEEE (2005)
- Xia, T., et al.: On defining affinity graph for spectral clustering through ranking on manifolds. *Neurocomputing* **72**(13–15), 3203–3211 (2009)
- Chauhan, S., Girvan, M., Ott, E.: Spectral properties of networks with community structure. *Phys. Rev. E* **80**(5), 056114 (2009)
- Arenas, A., Diaz-Guilera, A., Pérez-Vicente, C.J.: Synchronization reveals topological scales in complex networks. *Phys. Rev. Lett.* **96**(11), 114102 (2006)
- Cheng, X.-Q., Shen, H.-W.: Uncovering the community structure associated with the diffusion dynamics on networks. *J. Stat. Mech. Theory Exp.* **2010**(04), P04024 (2010)
- Newman, M.E.: Modularity and community structure in networks. *Proc. Natl. Acad. Sci.* **103**(23), 8577–8582 (2006)
- Shen, H.-W., Cheng, X.-Q., Fang, B.-X.: Covariance, correlation matrix, and the multiscale community structure of networks. *Phys. Rev. E* **82**(1), 016114 (2010)
- Zhang, X., You, Q.: An improved spectral clustering algorithm based on random walk. *Frontiers of Computer Science in China* **5**(3), 268 (2011)
- Ren, S., Zhang, S., Wu, T.: An improved spectral clustering community detection algorithm based on probability matrix. *Discret. Dyn. Nat. Soc.* **8**, 1–6 (2020)
- Hu, F., et al.: Community detection in complex networks using Node2vec with spectral clustering. *Physica A* **545**, 123633 (2020)
- Wang, Z., et al.: A community detection algorithm based on topology potential and spectral clustering. *Sci. World J.* (2014). <https://doi.org/10.1155/2014/329325>
- Mahmood, A., Small, M.: Subspace based network community detection using sparse linear coding. *IEEE Trans. Knowl. Data Eng.* **28**(3), 801–812 (2015)
- Steinhaeuser, K., Chawla, N.V.: Identifying and evaluating community structure in complex networks. *Pattern Recogn. Lett.* **31**(5), 413–421 (2010)
- Nawaz, W., et al.: Intra graph clustering using collaborative similarity measure. *Distrib. Parallel Datab.* **33**(4), 583–603 (2015)
- Zhou, H., et al.: A graph clustering method for community detection in complex networks. *Physica A* **469**, 551–562 (2017)
- Agrawal, S., Patel, A.: SAG Cluster: An unsupervised graph clustering based on collaborative similarity for community detection in complex networks. *Physica A* **563**, 125459 (2021)
- Ayoub, R.: Euler and the zeta function. *Am. Math. Mon.* **81**(10), 1067–1086 (1974)
- Li, W., Jiang, S., Jin, Q.: Overlap community detection using spectral algorithm based on node convergence degree. *Futur. Gener. Comput. Syst.* **79**, 408–416 (2018)
- Alinezhad, E., et al.: Community detection in attributed networks considering both structural and attribute similarities: two mathematical programming approaches. *Neural Comput. Appl.* **32**(8), 3203–3220 (2020)
- Zhou, Y., Cheng, H., Yu, J.X.: Graph clustering based on structural/attribute similarities. *Proc. VLDB Endow.* **2**(1), 718–729 (2009)
- Meng, F., et al.: Coupled node similarity learning for community detection in attributed networks. *Entropy* **20**(6), 471 (2018)
- Jia, C., et al.: Node attribute-enhanced community detection in complex networks. *Sci. Rep.* **7**(1), 1–15 (2017)
- Pizzuti, C., Socievole, A.: A genetic algorithm for community detection in attributed graphs. In: *International Conference on the Applications of Evolutionary Computation*. Springer (2018)
- Berahmand, K., et al.: A new attributed graph clustering by using label propagation in complex networks. *J. King Saud Univ. Comput. Inf. Sci.* (2020). <https://doi.org/10.1016/j.jksuci.2020.08.013>



42. Cheng, H., et al.: Clustering large attributed information networks: an efficient incremental computing approach. *Data Min. Knowl. Disc.* **25**(3), 450–477 (2012)
43. Huang, X., Cheng, H., Yu, J.X.: Dense community detection in multi-valued attributed networks. *Inf. Sci.* **314**, 77–99 (2015)
44. Gao, H., Huang, H.: Deep attributed network embedding. In: *Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)* (2018)
45. Cao, S., Lu, W., Xu, Q.: Grarep: learning graph representations with global structural information. In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management* (2015)
46. Tang, J., et al.: Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web* (2015)
47. Le, T.M., Lauw, H.W.: Probabilistic latent document network embedding. In: *2014 IEEE International Conference on Data Mining. IEEE* (2014)
48. Zhang, Z., et al.: ANRL: attributed network representation learning via deep neural networks. In: *IJCAI* (2018)
49. Wang, C., et al.: Attributed graph clustering: a deep attentional embedding approach. (2019). <http://arxiv.org/abs/1906.06532>
50. Zhang, X., et al.: Attributed graph clustering via adaptive graph convolution. (2019). <http://arxiv.org/abs/1906.01210>
51. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. (2016). <http://arxiv.org/abs/1609.02907>
52. Henaff, M., Bruna, J., LeCun, Y.: Deep convolutional networks on graph-structured data. (2015). <http://arxiv.org/abs/1506.05163>.
53. Wang, C., et al.: Mgae: marginalized graph autoencoder for graph clustering. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. (2017)
54. Sun, H., et al.: Network embedding for community detection in attributed networks. *ACM Trans. Knowl. Discov. Data (TKDD)* **14**(3), 1–25 (2020)
55. Luo, M., Yan, H.: Adaptive attributed network embedding for community detection. In: *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*. Springer (2020)
56. Zhou, Z., Amini, A.A.: Analysis of spectral clustering algorithms for community detection: the general bipartite setting. *J. Mach. Learn. Res.* **20**(47), 1–47 (2019)
57. Gulikers, L., Lelarge, M., Massoulié, L.: A spectral method for community detection in moderately sparse degree-corrected stochastic block models. *Adv. Appl. Probab.* **49**, 686–721 (2017)
58. Li, Y., et al.: Local spectral clustering for overlapping community detection. *ACM Trans. Knowl. Discov. Data (TKDD)* **12**(2), 1–27 (2018)
59. Liu, F., et al.: Global spectral clustering in dynamic networks. *Proc. Natl. Acad. Sci.* **115**(5), 927–932 (2018)
60. Ye, F., et al.: Homophily preserving community detection. *IEEE Trans. Neural Netw. Learn. Syst.* **31**(8), 2903–2915 (2019)
61. Nasiri, E., Berahmand, K., Li, Y.: A new link prediction in multiplex networks using topologically biased random walks. *J. Chaos Solit. Fract.* **151**, 111230 (2021)
62. Forouzandeh, S., Rostami, M., Berahmand, K.: Presentation a trust walker for rating prediction in recommender system with biased random walk: effects of H-index centrality, similarity in items and friends. *Eng. Appl. Artif. Intell.* **104**, 104325 (2021)
63. Berahmand, K., et al., *A preference random walk algorithm for link prediction through mutual influence nodes in complex networks*. *J. King Saud Univ. Comput. Inf. Sci.* (2021). <https://doi.org/10.1016/j.jksuci.2021.05.006>
64. Liu, W., Lü, L.: Link prediction based on local random walk. *EPL (Europhysics Letters)* **89**(5), 58007 (2010)
65. Kuncheva, L.I., Hadjitodorov, S.T.: Using diversity in cluster ensembles. In: *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*. IEEE (2004)
66. Rand, W.M.: Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.* **66**(336), 846–850 (1971)
67. Chen, M., Kuzmin, K., Szymanski, B.K.: Community detection via maximization of modularity and its variants. *IEEE Trans. Comput. Soc. Syst.* **1**(1), 46–65 (2014)
68. Pizzuti, C., Socievole, A.: Multiobjective optimization and local merge for clustering attributed graphs. *IEEE Trans. Cybernet.* **50**(12), 4997–5009 (2019)
69. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **78**(4), 046110 (2008)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



graph.

**Kamal Berahmand** received the M.S. degree in information technology engineering from Azarbaijan Shahid Madani University, Tabriz, Iran, in 2017. He is currently a Ph.D. student with the Faculty of Computer Engineering and Information Technology, Queensland University of Technology, Brisbane, Australia. His current research interests include Machine learning, network science, graph neural network, and knowledge



**Mehnoush Mohammadi** received the B.S. degree in computer engineering from the University of Kurdistan, Iran, in 2015, and the M.S degree in computer engineering, Artificial intelligence from the University of Kurdistan, Iran in 2018. Her research interest includes but is not limited to machine learning and complex network analysis.



**Azadeh Faroughi** has received her B.Sc. degree in Information Technology from Isfahan University in 2010. She has completed her M.Sc. and Ph.D. degrees in Information Technology, Computer Networks, from Sahand University of Technology in 2013 and Shiraz University of Technology in 2019 respectively. From 2018 to 2019, she was a visiting researcher at Polytechnic of Turin, where she worked on the application of machine learning

algorithms on internet traffic traces to mine information from logs generated by network elements. Her research interest includes, but is not limited to, machine learning, Big Data, computer networks, and traffic classification.



**Rojiar Pir Mohammadiani** received a Ph.D. degree in Information Technology, E-commerce from K.N.Toosi University of Technology, Tehran, in 2017. She joined the Department of Computer Engineering, the University of Kurdistan as an Assistant Professor in 2017. Her research interests include social network analysis, machine learning, and deep learning.